

4 Em Linha

Programação Avançada
Relatório
2020/2021 Meta 1

23 Maio

Luís Filipe Falcão Gomes nº 2016017431

Introdução

Neste trabalho foi desenvolvida uma aplicação em Java que simula o clássico jogo “4 em Linha” através de uma interface de texto.

Nesta versão estão incluídas:

Todas as regras do jogo;

Todas as funcionalidades requeridas pelo enunciado da disciplina.

Os ficheiros estão distribuídos por 5 packages:

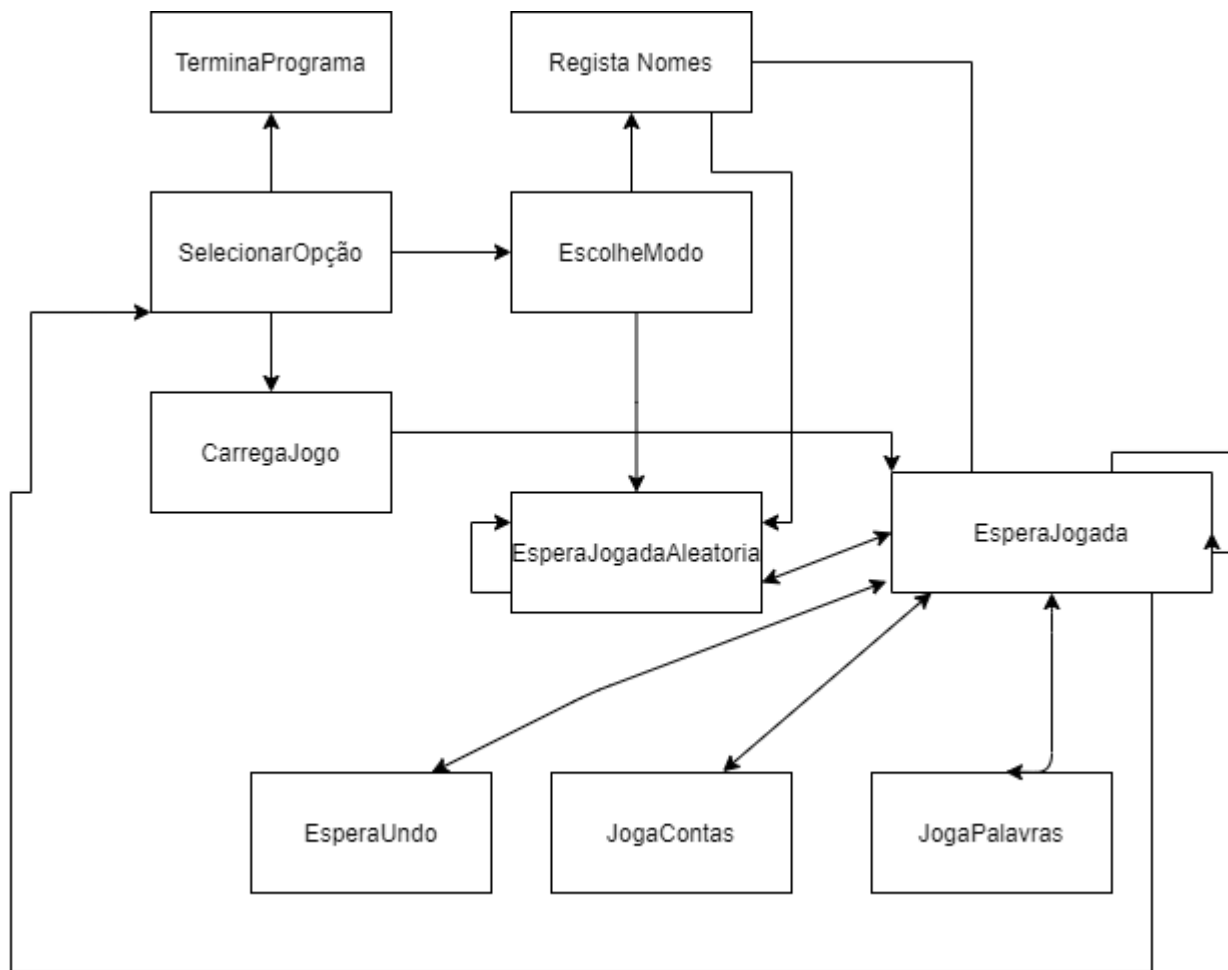
Jogo – package que abrange tudo;

Jogo.Logica- package que tem toda a lógica;

Jogo.Logica.Dados – package que contém os dados do jogo;

Jogo.Logica.Estados – package que contém os estados;

Jogo.ui – package que tem a interface;



ESTADOS:

SelecionaOpção:

O programa encontra-se neste estado enquanto estiver à espera do input por parte do utilizador anteriormente a iniciar o jogo. Nesta fase apenas temos três opções:

- *Iniciar Jogo : avança para o EscolheModo.*
- *Carregar Jogo : avança para o CarregaJogo.*
- *Terminar Jogo : avança para o TerminaJogo.*

CarregaJogo:

O programa encontra-se neste estado enquanto estiver á espera que o utilizador informe do ficheiro binário a carregar.

Este estado depois carrega o jogo e avança para o EsperaJogada.

TerminaPrograma:

Este estado termina o jogo.

EscolheModo:

O programa encontra-se neste estado enquanto o utilizador não escolher a quantidade de jogadores humanos, após o input tem dois caminhos que pode seguir:

Caso seja 1 ou 2 jogadores, avança para o RegistaNomes.

Caso seja 0, os jogadores são criado como bots e avança para o EsperaJogadaAleatória.

RegistaNomes:

Este estado espera que o/os utilizadores insiram os seus nomes antes de prosseguir para o esperaJogada.

EsperaJogada:

O EsperaJogada é o estado onde o jogo espera pela jogada do utilizador.

Este estado tem vários caminhos:

EsperaJogada: Pode voltar a si próprio caso sejam 2 jogadores humanos.

EsperaJogadaAleatória;

JogaContas : Minijogo nº1;

JogaPalavras : Minijogo nº2;

EsperaUndo;

SelecionaOpção : Caso o jogo acabe.

EsperaJogadaAleatória:

Este estado espera que o jogo calcule a próxima jogada que os jogadores bots vão jogar.

JogaPalavras:

O programa entra neste estado quando o utilizador estiver no Jogo das Palavras.

JogaContas:

O programa entra neste estado quando o utilizador estiver no Jogo das Contas.

EsperaUndo:

Este é o estado onde o jogo espera que o utilizador informe quantas jogadas quer voltar atrás.

CLASSES:

Main:

Apenas possui a criação e início da interface gráfica; Apenas possui a criação e início da interface gráfica;

Interface:

É a classe que possui a interface do jogo.

IStates:

Classe interface;

StateAdapter:

Classe que implementa a interface "IStates";

DadosJogo:

Esta classe serve como banco de dados para todo o jogo, onde os mesmos podem ser acedidos e modificados;

Jogo:

Classe que contém os dados e os estados do jogo;

JogoContas:

Classe que contém e implementa os métodos do jogo das contas;

JogoPalavras:

Classe que contém e implementa os métodos do jogo das palavras;

Caretaker:

Classe que guarda e interpreta os mementos do jogo;

Memento:

Classe que guarda os snapshots que recebe do tabuleiro;

Player:

Classe que contém os dados sobre os jogadores;

Tabuleiro:

Classe que contém os dados sobre o tabuleiro e que desempenha um papel de originator, pois ele cria os seus próprios snapshots,

Imemento:

Classe interface dos mementos

Operadores:

Classe Enum que contém os operadores do jogo das Contas.

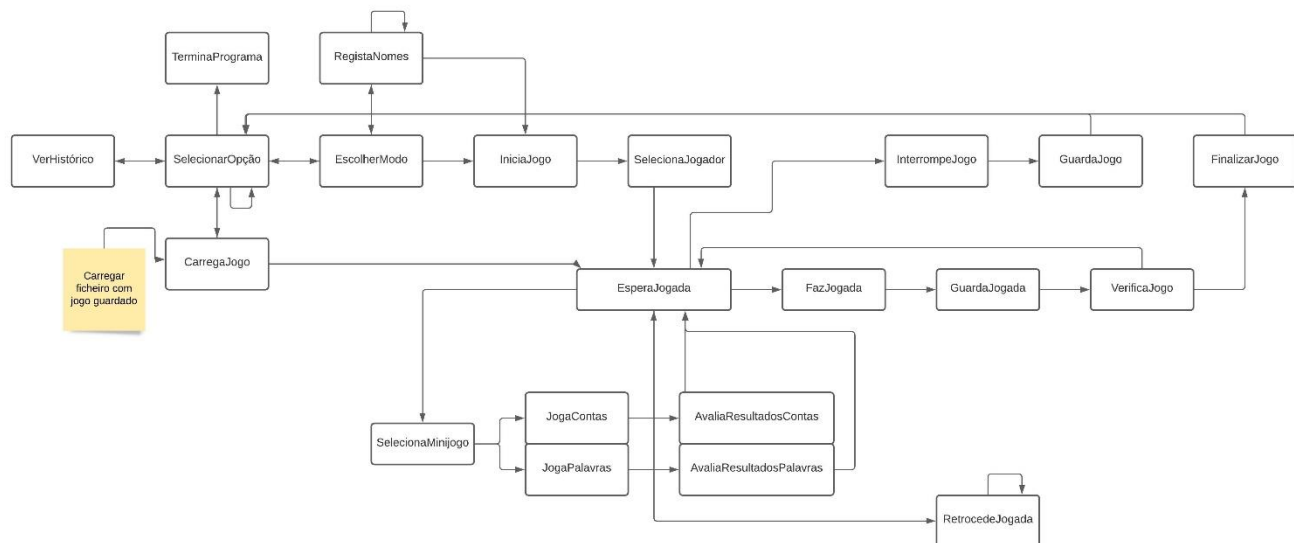
Opções de implementação:

Ao longo do trabalho, foram havendo várias versões com lógicas diferentes sendo que optei pela que apresento:

Classe Jogo: Antes a minha classe DadosJogos continha também os métodos de transição de estados, isso chegou me a causar muitos erros de Exceções. Ao criar a classe Jogo, que contém a chamada das funções dos DadosJogo e os métodos dos estados, resolvi muitos dos problemas das exceções.

Redução de estados:

Comecei o trabalho com cerca de 21 estados:



Com o decorrer do trabalho, fui reduzindo os estados pois quando ao utiliza-los reparei que eram redundantes e apenas tornavam o programa mais confuso e difícil de perceber.

